Estimation of Remaining Useful Life Based on Switching Kalman Filter Neural Network Ensemble

Pin Lim[1], Chi Keong Goh[2], Kay Chen Tan[3] and Partha Dutta[4]

[1,2,4] *Rolls Royce Singapore, Singapore*
Pin.Lim@Rolls-Royce.com
ChiKeong.Goh@Rolls-Royce.com
Partha.Dutta@Rolls-Royce.com

[3] *National University of Singapore, Singapore*
eletankc@nus.edu.sg

ABSTRACT

The proposed method is an extension of an existing Kalman Filter (KF) ensemble method. While the original method has shown great promise in the earlier PHM 2008 Data Challenge, the main limitation of the KF ensemble is that it is only applicable to linear models. In prognostics, degradation of mechanical systems is typically non-linear in nature, therefore limiting the applications of KF ensemble in this area. To circumvent this problem, this paper propose to approximate non-linear functions with piecewise linear functions. When estimating the RUL, the Switching Kalman Filter (SKF) is able to choose the most probable degradation mode and thus make better predictions. The implementation of the proposed SKF ensemble method is illustrated by implementing on NASA's C-MAPSS Dataset as well as the PHM 2008 Data Challenge Dataset. The results show the effectiveness of the SKF in detecting the switching point between various degradation modes as well as the improved accuracy of the SKF ensemble method compared to other available methods in literature.

1. INTRODUCTION

In the recent years, Condition Based Maintenance (CBM) has been garnering more attention as it allows industries to better plan logistics as well as save cost by replacing parts only when needed. Prognostics being one of the key enablers of CBM has therefore also gained more interest in both academia and industry. The key notion of prognostics, albeit not the only one, is to determine the time remaining before a likely failure. This value is commonly termed as the Remaining Useful Life (RUL) of the system.

In this paper, a novel prediction algorithm is presented which is applicable to non linear degradation models. The algorithm assumes that degradation model can be described by a number of piece-wise linear functions. With each of these linear functions describing a linear model, the most suitable model to describe the degradation at any point in time is chosen based on the Switching Kalman Filter (SKF) algorithm. The remainder of this paper is structured as follows, Section 2 first introduces the datasets used to evaluate the effectiveness of the algorithm. Section 3 follows by presenting a simple single neural network approach to evaluate the difficulty of the problem. Finally in Section 4 the SKF ensemble approach is presented and evaluated.

2. DATASET

In this paper a total of two datasets were used. The datasets used are namely the PHM 2008 Data Challenge Dataset as well as the NASA C-MAPSS Dataset (Saxena & Goebel, 2008), the C-MAPSS dataset is further divided into 4 sub-datasets as shown in Table 1. Both datasets contain simulated data produced using a model based simulation program (named Commercial Modular Aero-Propulsion System Simulation, C-MAPSS) developed by NASA (Saxena, Goebel, Simon, & Eklund, 2008).

Table 1. Dataset details (Simulated from C-MAPSS)

	C-MAPSS				**PHM**
Dataset	**FD001**	**FD002**	**FD003**	**FD004**	**2008**
Train Trajectories	100	260	100	248	218
Test Trajectories	100	259	100	248	218
Conditions	1	6	1	6	6
Fault Modes	1	1	2	2	2

The data is arranged in an n-by-26 matrix where n corresponds to the number of data points in each dataset. Each row is a snapshot of data taken during a single operational cycle and each column represents a different variable. Included in the data are three operational settings that have a substantial effect on engine performance.

Each trajectory within the train and test trajectories is assumed to the be life-cycle of an engine. While each engine is simulated with different initial conditions, these conditions are considered to be of normal conditions (no faults). For each engine trajectory within the training sets, the last data entry corresponds to the moment the engine is declared unhealthy. On the other hand the test sets terminate at some time prior to failure and the aim is to predict the number of Remaining Useful Life (RUL) of each engine of the test set.

For each of the C-MAPSS dataset the actual RUL value of the test trajectories were made available to the public while the actual RUL of the test dataset of PHM 2008 is not available. However, users can submit their results to the NASA website to obtain a score limited to one submission per day. Due to this constrain, most of the analysis done in this paper will be based on the NASA C-MAPSS dataset instead of the PHM 2008 dataset. The PHM 2008 dataset would instead be used for comparison against other algorithms proposed in literature.

2.1. Evaluation Metrics

2.1.1. Scoring Function

The scoring function used in this paper is identical to that used in PHM 2008 Data Challenge. This scoring function is illustrated in Eq. (1), where s is the computed score, N is the number of engines, and d = $\bar{R}UL$-RUL (Estimated RUL- True RUL).

$$s = \sum_{i=1}^{N} s_i, s_i = \begin{cases} \mathrm{e}^{-\frac{d_i}{13}} - 1 & \text{for } d_i < 0 \\ \mathrm{e}^{\frac{d_i}{10}} - 1 & \text{for } d_i \geq 0 \end{cases} \tag{1}$$

The characteristic of this scoring function is that it favours early predictions more than late predictions. This is in line with the risk adverse attitude in aerospace industries. However there are several drawbacks with this function. The most significant drawback being a single outlier would dominate the overall score, thus masking the true accuracy of the algorithm. Another drawback is the lack of consideration of the prognostic horizon of the algorithm. The prognostic horizon assess the time before failure which the algorithm is able to accurately estimate the RUL value within a certain confidence level. Finally this scoring function favours algorithms which artificially lowers the score by underestimating the RUL. Despite all these shortcomings, this scoring function is still used in this paper in order to provide a level comparison with other methods in literature.

2.1.2. RMSE

In addition to the scoring function, the Root Mean Square Error (RMSE) of the estimated RULs is also used as a performance measure. RMSE is chosen as it gives equal weight to both early and late predictions. Using RMSE in conjunction with the scoring function would prevent the user from favouring an algorithm which artificially lowers the score by underestimating but resulting in higher RMSE. Furthermore, various literature working on this dataset uses RMSE to evaluate their algorithms, inclusion of RMSE would therefore allow the author to compare results with those available in literature.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N} d_i^2} \tag{2}$$

A comparative plot between the two evaluation metrics is shown in Figure 1. It can be observed that at lower absolute error values the scoring function results in lower values than the RMSE. The relative characteristics of the two evaluation metrics will be useful during the discussion of results in the latter part of this paper.

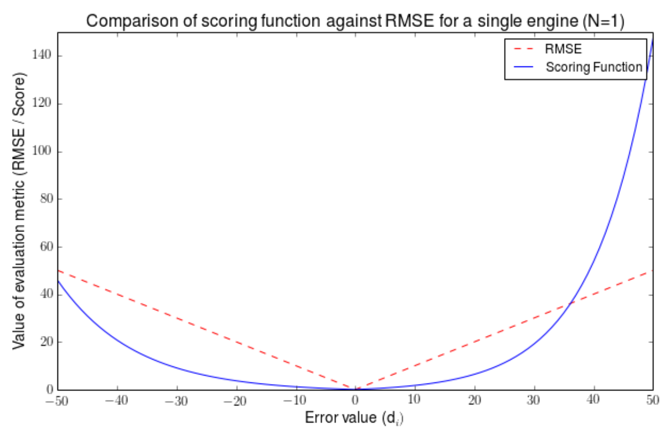

Figure 1. Comparison of evaluation metric values for different error values

2.2. Data Preparation

2.2.1. Operating Conditions

Several literature (Wang et al., 2008; Peel, 2008; Heimes, 2008), have shown that by plotting the operational setting values, the data points are clustered into six different distinct clusters. This observation is only applicable for datasets with different operational conditions, data points from FD001 and FD003 are all clustered at a single point instead. These clusters are assumed to correspond to the six different oper-

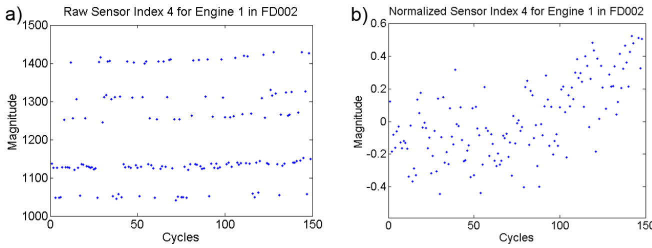

Figure 2. Sensor values (a) before and (b) after normalization

ational conditions. It is therefore possible include the operational condition history as a feature. This is done by adding 6 columns of data representing the number of cycles spent in their respective operational condition since the beginning of the series (Peel, 2008).

2.2.2. Data Normalization

Due to the 6 operating conditions, each of these operating conditions results in disparate sensor values as shown in Figure 2. Therefore prior to any testing and training, it is imperative to normalize the data points to be within the range of [-1,1] using Eq. (3). As normalization was carried out within the range of values for each sensor and each operating condition, this will ensure equal contribution from all features across all operating conditions (Peel, 2008). Alternatively, it is also possible to incorporate operating condition information within the data to take into consideration various operating conditions

$$Norm(x^{(c,f)}) = 2\frac{(x - x_{min}^{(c,f)})}{x_{max}^{(c,f)} - x_{min}^{(c,f)}} - 1, \forall c, f \qquad (3)$$

where c represents the operating conditions and f represents each of the original 21 sensors.

3. SINGLE NEURAL NETWORK APPROACH

3.1. Method Description

The aim of this section is two-fold. Firstly as a prior to experimenting with other methods, the complexity of the problem was tested using a single Multi-Layer Perceptron (MLP) Network to achieve a baseline performance. This baseline performance then used for comparing the accuracy of the proposed method. Secondly, the method is used to evaluate the performance of the two different RUL functions presented in section 3.2 below.

3.2. Arbitrary RUL Function

In its crudest form prognostic algorithms are similar to regression problems. However, unlike typical regression problems, an inherent challenge for data driven prognostic problems is determining the desired output values for each input data point. This is because in real world applications, it is impossible to accurately determine the health of the system at each time step without an accurate physics based model. A sensible solution would be to simply assign the desired output as the actual time left before functional failure (Peel, 2008; Baraldi, Mangili, & Zio, 2012). This approach however inadvertently implies that the health of the system degrades linearly with usage (Figure 3a).

An alternative approach is to derive the desired output values based on a suitable degradation model. For this data-set (Heimes, 2008) has proposed a piece-wise linear degradation model which limits the maximum value of the RUL function (Figure 3b). The maximum value was chosen based on the observations of the data and its numerical value is different for each data-set. For the sake of simplicity, the former will be addressed as 'linear function' while the latter will be known as the 'kink function' in the remainder of the paper.

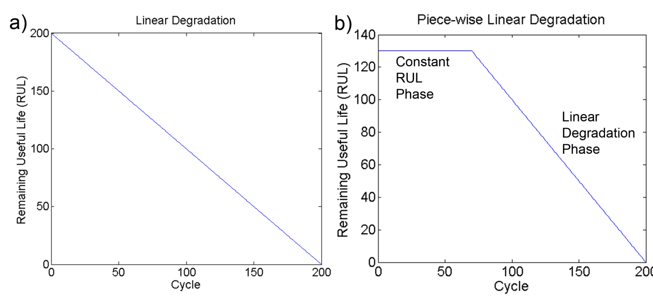

Figure 3. Comparison of degradation models. a) Linear Degradation model, b) Piece-wise Linear Degradation Model

Each of these approaches has their own advantages. The latter case is more likely to prevent the neural network from overestimating the RUL, it is also a more logical model as the degradation of the system typically only starts after a certain degree of usage. On the other hand, the former case follows the definition of RUL in the strictest sense which defined as the time to failure. Therefore the plot of time left of a system against the time passed naturally results in a the linear function as shown in Figure 3a. However it should be noted that in cases where knowledge of a suitable degradation model is unavailable, the linear model is the most natural choice to use.

3.3. Results

For each sub-dataset within the C-MAPSS dataset, two MLPs were individually trained using the linear and kink RUL functions as desired outputs. The MLPs were then tested using the corresponding test sub-datasets and evaluated using Eq. (1) and Eq. (2). Due to the inherent noise in the data, in order to capture the variance of each MLP, the whole training and testing process was repeated for a total of 10 trials. The results from these trials are expressed in the form of box plots shown in Figure 4 & 5.

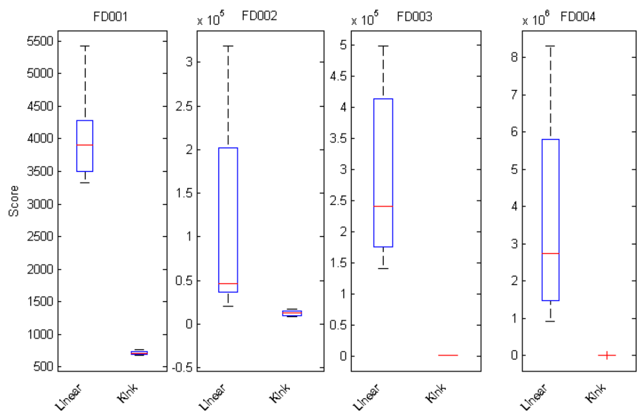

Figure 4. Scores of MLP trained with linear and kink RUL functions.

Figure 4 shows that using the linear RUL function resulted in comparatively much higher variance in scores. However considering the RMSE plots (Figure 5) the variance of RMSE values within each dataset is relatively similar. Therefore the higher variance in scores is due to the nature of the scoring function. The exponential term in the scoring function could cause large deviations in the score due to a single inaccurate estimation. The variance of the RMSE values for both MLPs could be attributed to the inability of the single MLP to handle noisy input data.

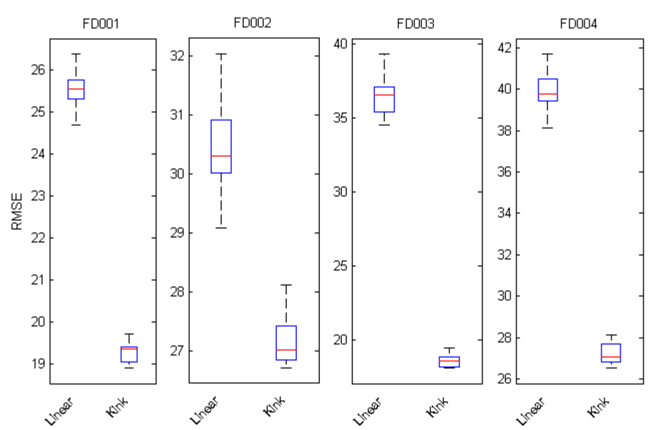

Figure 5. RMSE of MLP trained with different RUL functions.

More importantly, all datasets show significant improvements in both RMSE and scores when the kink RUL function is used. The lower RMSE values obtained by using the kink RUL function (Figure 5) is evidence that their respective lower scores in Figure 4 is due to more accurate predictions instead of inducing underestimation of RUL. These results agree with Heimes (2008) that the kink RUL function is a much more suitable degradation model for these datasets.

4. SWITCHING KALMAN FILTER (SKF) ENSEMBLE

4.1. Method Description

In order to improve the prognostic accuracy of a single MLP implemented in section 3.3, ensemble methods are explored to develop a more accurate and robust prognostic method. Ensemble methods are generally used to combine multiple weak classifiers into a single strong classifier. It has been found that ensembles would have higher accuracy and generalizability if each ensemble members are accurate and make errors on different parts of the input space (Maclin & Opitz, 2011). There are generally two main steps in creating an ensemble: The first step is to create individual ensemble members, and the second step to combine the output of the ensemble members.

In order for the ensemble to generate better results, the generalization of the ensemble must be improved. This can be obtained by having diversity in the ensemble members. The most commonly used method to create ensemble members include input data sampling techniques such as Bagging and Boosting (Zhou, 2012; Re & Valentini, 2011). In this paper, networks with different network topology are used to create ensemble members as this method has less variables to tune as compared to boosting and bagging.

Combination of output from ensemble members is usually taken as a weighted mean or median of the ensemble member outputs (Zhou, 2012). The weights are usually determined based on the training error of each ensemble member (Krogh & Vedelsby, 1995). Peel (2008) proposed an alternative combination method which uses a Kalman filter to combine the output of several neural networks. This method has shown great promise by wining the IEEE Gold for PHM 2008 Data Challenge. In his work, both the training function for the neural networks and the model used in the Kalman filter assumes a linear degradation function thus limiting its application to linear cases. This section extends this method by using a Switching Kalman Filter (SKF) for piecewise linear applications. Thus allowing implementation of a similar ensemble for other degradation patterns.

4.2. Ensemble Members

In this paper MLPs with different number of hidden neurons are used as ensemble members. The number of hidden neurons were randomly picked from a uniform distribution of integers between 5 to 25 inclusive. The maximum number of hidden neurons was limited to prevent over fitting on the training set, thus ensuring generalization on unseen data points. A total of 4 ensemble members were generated per ensemble.

4.3. Aggregation based on Kalman Filter (KF)

KFs and its variants have been widely used for machine learning applications. These applications range from simple state

prediction (Borguet & Léonard, 2009) to training of neural network weights using the Extended Kalman Filter (EKF) (Singhal & Wu, 1989; Puskorius & Feldkamp, 1991). In this paper, the traditional KF and its variant the SKF will be used.

4.3.1. Kalman Filter

The more commonly used application of the KF is as a forward pass state estimator. The filter predicts the hidden states for the next time step given the history of estimated states and observing noisy outputs. The predicted states are considered optimal as the filter aims to minimize the uncertainties in the estimate (AL-Mathami, Everson, & Fieldsend, 2012). Prior to using the KF, the system must be modeled as a linear system as shown

$$\begin{aligned} x_t &= Ax_{t-1} + w_t \\ z_t &= Hx_t + v_t \end{aligned} \tag{4}$$

where x_t is the state vector at time t, A is the transition matrix, z_t represents the output observations, H is the observation matrix, w_t and v_t are the process noise and observation noise respectively. Based on the model a recursive process is then carried out whereby the prediction step is carried out by

$$\begin{aligned} \hat{x}_t &= A\bar{x}_{t-1} \\ \hat{P}_t &= A\bar{P}_{t-1}A^T + Q \end{aligned} \tag{5}$$

where P_t is the state covariance matrix and Q is the process error covariance matrix. The KF then updates the estimate based on the new observations. The updating step is then carried out by the following equations

$$\begin{aligned} K_t &= \hat{P}_t H^T [H\hat{P}_t H^T + R]^{-1} \\ \bar{x}_t &= \hat{x}_t + K_t[z_t - H\hat{x}_t] \\ \bar{P}_t &= [I - K_t H]\hat{P}_t \end{aligned} \tag{6}$$

where R is the observation error covariance matrix and K_t is the Kalman gain at time t. For illustrative purposes, the state x_t is chosen as

$$x_t = \begin{bmatrix} RUL_t \\ \Delta RUL_t \end{bmatrix}, \Delta RUL_t = RUL_t - RUL_{t-1} \tag{7}$$

It is therefore straight forward to express the kink RUL function as a piecewise linear function with their respective linear KF model expressed as

$$A_c = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \; A_l = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \tag{8}$$

where A_c is the model for the initial constant RUL phase and A_l is the model for the linear degradation phase, assuming a gradient of -1 for the linear degradation phase. In addition, the outputs from individual neural networks are taken to be the observations, therefore the observation vector z_t and H are set as

$$z_t = \begin{bmatrix} R\hat{U}L_1 \\ \dots \\ R\hat{U}L_n \end{bmatrix}, H = \begin{bmatrix} 1 & 0 \\ \dots & \dots \\ 1 & 0 \end{bmatrix} \tag{9}$$

where $R\hat{U}L_n$ is the output of the n^{th} neural network in the ensemble. Further details of modeling the ensemble outputs is covered in Peel (2008) and Baraldi et al. (2012).

4.3.2. Kalman Smoother

In contrast to the KF, which estimates the optimal state given observations up to time t, the Kalman smoother aims to estimate the optimal state at time t given the observations from 1 to T, where T represents the total length of data observations (AL-Mathami et al., 2012). The Kalman smoother is an analogous backwards recursive process which estimates the states from the end of the observation data. Therefore combining both forward and backward pass gives the optimal estimated state given the whole observation data.

At the last time step the variables $\tilde{x}$ and $\tilde{P}$ are initialized as

$$\begin{aligned} \tilde{x}_T &= \bar{x}_T \\ \tilde{P}_T &= \bar{P}_T \end{aligned} \tag{10}$$

where $\tilde{x}$ is the smoothed state and $\tilde{P}$ is the smoothed covariance. The smoothed states can then be calculated based on the following recursive equations where t decreases from $T-1$ to 1 (AL-Mathami et al., 2012).

$$\begin{aligned} J_t &= (\bar{P}_t A^\top)\hat{P}_{t+1}^{-1} \\ \tilde{x}_t &= \bar{x}_t + (J_t(\tilde{x}_{t+1}^\top - A\tilde{x}_t^\top))^\top \\ \tilde{P}_t &= \bar{P}_t + J_t(\tilde{P}_{t+1} - \hat{P}_{t+1})J_t^\top \end{aligned} \tag{11}$$

4.4. Switching Kalman Filter (SKF)

Eq. (8) in the earlier section has shown that the Kink degradation function can be modeled using two linear systems. The outputs of the ensemble members would therefore need to be combined using the suitable KF model. This problem is further compounded by the fact that the switching point between the two models differ for every engine. Thus making it difficult to pre-define a rule to switch between the two models. To circumvent this problem a SKF (Murphy, 1998; AL-Mathami et al., 2012) is implemented to autonomously determine the switching point.

In this application, SKF predicts the most probable hidden discrete model given the observations and the models. The

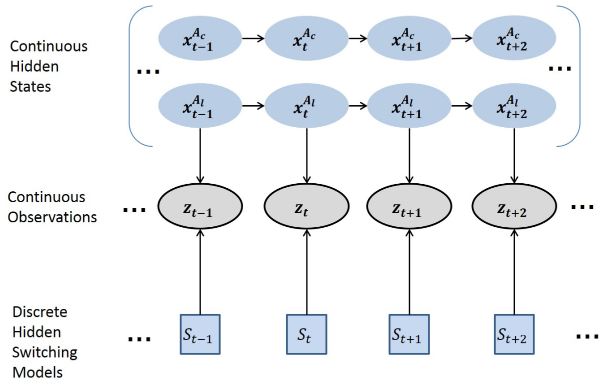

Figure 6. Directed graphical probabilistic model of SKF

graphical probabilistic model of the SKF for aggregating ensemble methods is shown in Figure 6. Based on the figure, the SKF determines the sequence of models which would most likely result in the series of observations. Similar to the KF, the SKF computes the posterior probability of the model given the observations in two passes. The forward pass calculates $P(S_t = j|x_t, x_{1:t-1})$ while the backwards pass calculates $P(S_t = j|x_{t:T})$. An illustrative example of the forward pass calculation is shown below

For each t, j:

$$P(S_t = j|x_t, x_{1:t-1}) = \frac{P(x_t|S_t=j,x_{1:t-1})P(S_t=j|x_{1:t-1})}{P(x_t|x_{1:t-1})}$$
$$= \frac{1}{c}L_t(j)\Sigma_i Z(i,j)P(S_{t-1} = i|x_{1:t-1}) \quad (12)$$

where

$$c = P(x_t|x_{1:t-1}) = \Sigma_j L_t(j)\Sigma_i Z(i,j)P(S_{t-1} = i|x_{1:t-1})$$
$$L_t = P(x_t|S_t = j, x_{1:t-1}) \sim N(x_t, A_j x_{t-1}, Q_j)$$
$$Z(i,j) = P(S_t = j|S_t = i, x_{1:t-1}) \quad (13)$$

It should be noted that $Z(i, j)$ is a predefined transition matrix which contains the probability of transition from one model to another. Thus, based on this calculated probability, the most probable model can be chosen. The backwards pass can be calculated in a similar manner and therefore will not be repeated here. For more details on the SKF, readers can refer to Murphy (1998) and AL-Mathami et al. (2012)

In this implementation, the output of the trained ensemble members are taken to be the observations and switching models corresponds to the two KF models expressed in Eq. (8). The most probable sequence of models is first determined by the SKF, the corresponding KF models can then be applied to aggregate the outputs of individual ensemble members to obtain the estimated RUL value. Figure 7 shows an example of the SKF algorithm estimating the degradation of an engine from the training set. It can be observed that the predicted switching point between the two models by the SKF corresponds well with the predefined kink location in the RUL function. It should also be noted that the initial conditions of the Kalman filter is re-initialized for each engine.

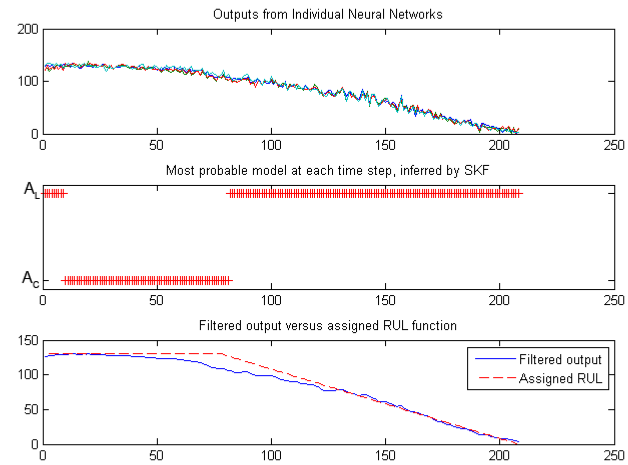

Figure 7. Example of SKF Ensemble output on a training engine

4.5. Results

In this section the performance of the SKF ensemble is illustrated and compared with the original KF ensemble method. The KF ensemble was recreated to the best of knowledge based on the details given in Peel (2008). Furthermore, results obtained from Section 3.3 are also included for comparison purposes to highlight the effectiveness of ensemble methods. Similar to previous sections, all the experiments were repeated for a total of 10 trials, the results obtained from these trials are then expressed in the form of a boxplot.

4.5.1. C-MAPSS Dataset

Figure 8 illustrates the scores of all methods described in this paper for all four sub-datasets within C-MAPSS. It is observed that both linear MLP or KF ensemble displayed high mean and large variance of scores. In addition all four methods achieved RMSE values of the same order (Figure 9). Based on these observations, coupled with the characteristics of each evaluation metric (Figure 1), it can be implied that the high scores are caused by certain outliers in predicting the RUL. This phenomenon could probably be attributed to the use of the linear RUL function which might lead to overestimating of the RUL, thus resulting in significantly higher scores.

In addition, the high scores exhibited by the Linear MLP and KF ensemble resulted in a badly scaled boxplot making it difficult to illustrate and compare the relative performance of the remaining algorithms. Therefore more in depth comparison of the four methods will focus mainly on the RMSE values instead (Figure 9).

Based on Figure 9, it can also be deduced that the SKF ensemble outperforms that KF ensemble significantly. The SKF ensemble achieved much lower RMSE values which is most likely attributed to the use of the kink RUL function to model the degradation of the system. These results reaffirm the hypothesis arrived in Section 3.3 that the kink RUL function is a much more accurate model for this dataset.

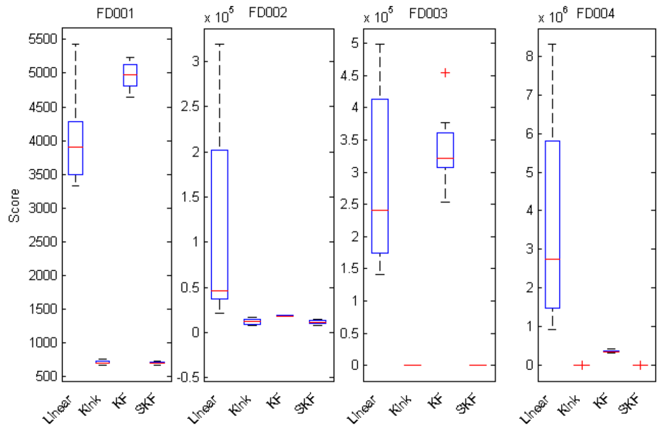

Figure 8. Scores of various algorithms for all C-MAPSS Datasets.

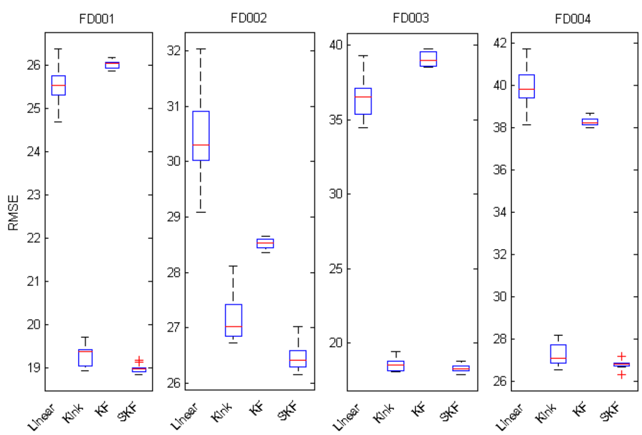

Figure 9. RMSE of various algorithms for all C-MAPSS Datasets.

As expected, both KF and SKF ensemble methods resulted in significantly lower RMSE variance compared to their respective linear and kink MLPs. This can be attributed to the ability of ensembles to aggregate the outputs of individual ensemble members thus resulting in a lower variance. In addition, the use of KF helps to filter out noise from the output of the ensemble (Figure 7) thus resulting in increased robustness against inherent noise in the data. The same observations can be seen in Figure 10 which shows in greater detail the comparison box plot between the SKF ensemble and the single MLP trained with a kink training function. In addition to obtaining lower variance in RMSE values, the SKF ensemble also exhibited lower mean RMSE values. Thus showing that the SKF ensemble outperforms the original MLP in both accuracy and variance in predictions.

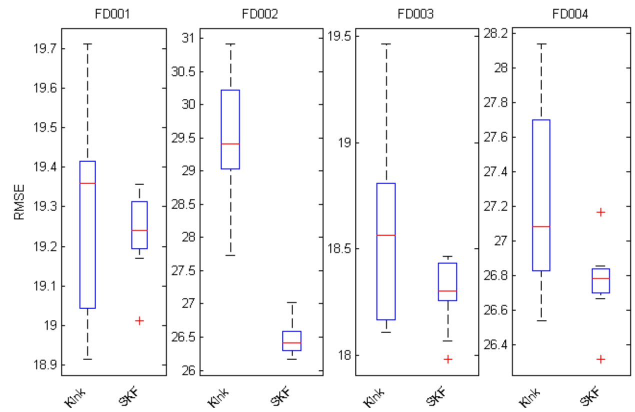

Figure 10. RMSE of MLP with Kink training function and SKF for all C-MAPSS Datasets.

Comparing the scores between the Kink MLP and the SKF ensemble (Figure 11) for all datasets showed that both methods achieved scores within the similar range. However the SKF slightly out performs the Kink MLP by exhibiting less variance in scores throughout the 10 trials. This phenomenon can be similarly be attributed the ability of ensemble to be more robust to noise as mentioned in the earlier paragraph.

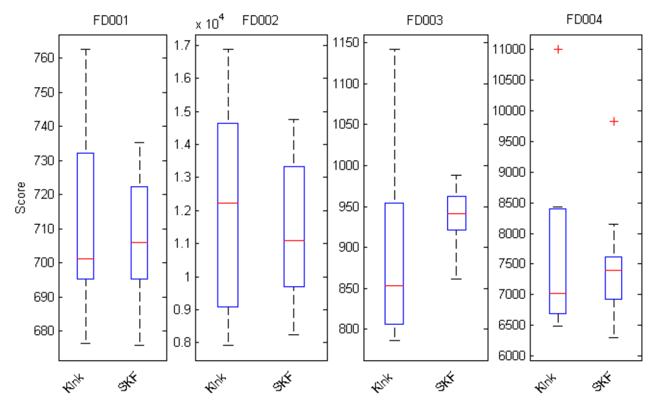

Figure 11. Scores of MLP with Kink training function and SKF for all C-MAPSS Datasets.

4.5.2. PHM 2008 Dataset

In this section, the algorithms were tested on the test dataset for PHM 2008. The estimated RULs of 218 engines within the dataset were then uploaded to the NASA Data Repository website and a single score was then returned by the website. The results were also compared with available literature that provided suitable scores for comparison.

Based on the results it can be seen that the SKF ensemble produces significantly lower scores and outperforms the other

Table 2. Scores for various algorithms on PHM 2008 test dataset

Methods	Scores
Single MLP (Linear)	118338
Single MLP (Kink)	6103.46
KF Ensemble	5590.03
SKF Ensemble	2922.33
Gibbs Filtering (Le Son, Fouladirad, & Barros, 2012)	4170

methods. However as mentioned in Section 2, submission of estimated RULs are limited to once a day. Thus the scores shown in Table 3 are from a single submission. Therefore these scores are also subject to variance as seen in earlier sections.

5. CONCLUSION

In this paper we have demonstrated the effectiveness of a SKF ensemble for systems with non-linear degradation patterns. In addition, the performance of the SKF ensmeble on NASA's C-MAPSS dataset has shown improvement over other methods in literature. Implementation on these simulated datasets simply serve as a proof-of-concept for the proposed method at this stage. This method has also wide applications to other prognostic situations where the system involved has more than one degradation mode. An example would be where the degradation pattern of the system changes due to external factors such as operating conditions or overhaul maintenance. In view of the range of possible applications, the authors have plans to implement the proposed method on a real-world dataset and validate its effectiveness.

ACKNOWLEDGMENT

This work was supported by the Economic Development Board Industrial Postgraduate Programme Fund under the project R-263-000-A14-592.

REFERENCES

AL-Mathami, Y. S., Everson, R., & Fieldsend, J. (2012). *Probabilistic controllled airpsace infringement tool.* Unpublished doctoral dissertation.

Baraldi, P., Mangili, F., & Zio, E. (2012). A kalman filter-based ensemble approach with application to turbine creep prognostics. *IEEE TRANSACTIONS ON RELIABILITY*, *61*.

Borguet, S., & Léonard, O. (2009). Coupling principal component analysis and kalman filtering algorithms for on-line aircraft engine diagnostics. *Control Engineering Practice*, *17*(4), 494–502.

Heimes, F. (2008). Recurrent neural networks for remaining useful life estimation. In *Prognostics and health management, 2008. phm 2008. international conference on* (pp. 1–6).

Krogh, A., & Vedelsby, J. (1995). Neural network ensembles, cross validation, and active learning. *Advances in neural information processing systems*, 231–238.

Le Son, K., Fouladirad, M., & Barros, A. (2012). Remaining useful life estimation on the non-homogenous gamma with noise deterioration based on gibbs filtering: A case study. In *Prognostics and health management (phm), 2012 ieee conference on* (pp. 1–6).

Maclin, R., & Opitz, D. (2011). Popular ensemble methods: An empirical study. *arXiv preprint arXiv:1106.0257*.

Murphy, K. P. (1998). *Switching kalman filters* (Tech. Rep.). Citeseer.

Peel, L. (2008). Data driven prognostics using a kalman filter ensemble of neural network models. In *Prognostics and health management, 2008. phm 2008. international conference on* (pp. 1–6).

Puskorius, G. V., & Feldkamp, L. A. (1991). Decoupled extended kalman filter training of feedforward layered networks. In *Neural networks, 1991., ijcnn-91-seattle international joint conference on* (Vol. 1, pp. 771–777).

Re, M., & Valentini, G. (2011). Ensemble methods: a review.

Saxena, A., & Goebel, K. (2008). *Phm08 challenge data set, nasa ames prognostics data repository.* Moffett Field, CA. Retrieved from [http://ti.arc.nasa.gov/project/prognostic-data-repository]

Saxena, A., Goebel, K., Simon, D., & Eklund, N. (2008). Damage propagation modeling for aircraft engine run-to-failure simulation. In *Prognostics and health management, 2008. phm 2008. international conference on* (pp. 1–9).

Singhal, S., & Wu, L. (1989). Training feed-forward networks with the extended kalman algorithm. In *Acoustics, speech, and signal processing, 1989. icassp-89., 1989 international conference on* (pp. 1187–1190).

Wang, T., Yu, J., Siegel, D., & Lee, J. (2008). A similarity-based prognostics approach for remaining useful life estimation of engineered systems. In *Prognostics and health management, 2008. phm 2008. international conference on* (pp. 1–6).

Zhou, Z.-H. (2012). *Ensemble methods: foundations and algorithms*. CRC Press.